UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/089,139	08/19/2002	Adam Bosworth	41016.P009	2275

25943 7590 03/22/2007

SCHWABE, WILLIAMSON & WYATT, P.C.
PACWEST CENTER, SUITE 1900
1211 SW FIFTH AVENUE
PORTLAND, OR 97204

EXAMINER	
RAMPURIA, SATISH	
ART UNIT	**PAPER NUMBER**
2191	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/089,139	BOSWORTH ET AL.	
	Examiner	Art Unit	
	Satish S. Rampuria	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 December 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Response to Amendment

1. This action is in response to the Amendment received on December 27, 2006.

2. The specification due to use of the trademark "Java" has been withdrawn in view of applicants amendment to specification.

3. Claims 4, 5, 8, 23, 24 and 27 were rejected under 35 U.S.C. 112, second paragraph, due to use of the trademark "Java" has been withdrawn in view of applicants amendment to claims.

4. Claims amended by the Applicants: 1, 4-5, 8, 14, 17, 19-20, 23-24, 27, 33, 36, and 38.

5. Claims pending in the application: 1-38.

Response to Arguments

6. Applicant's arguments with respect to claims have been considered but they are not persuasive.

In the remarks, the applicant has argued that:

a. Even if it were possible to read the Java VM and VB Script Interpreter as first and second code statement processing units, as the Examiner does (which it is not, for reasons given below), Wang does not disclose an execution engine that invokes **both** of the Java VM and VB Script Interpreter. The HTML parser of Wang, described above, simply creates intermediate sources and does not invoke either of the Java VM or the VB Script Interpreter. According to Wang, col. 3, lines 57-67 and col. 4, lines 1-8, the Java VM is invoked first at runtime, and the VB Script Interpreter is later invoked by the Java VM. Thus, there is no common process that invokes both the Java VM and the VB Script

Interpreter, and therefore, Wang does not disclose the execution engine recited by claim 1.

Examiner's response:

In response to Applicants argument, Wang discloses enabling multiple runtime processor executed by the computer. Each of the runtime processors process their respective intermediate sources derived from an original input source, i.e., Java or Visual Basic Script (See summary). In order to process multi language processor, Wang's system recognize different input source languages and invokes the respective processor according to the input source language (col. 2, lines 26-35). Further, Wang's system has a parser, which recognizes the input sources language, and sends to the appropriate translator (col. 3, lines 24-30 and FIG. 2). Furthermore, applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., no common process) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

b. Additionally, even assuming Wang anticipates claims 1 and 6, and/or claims 20 and 25, Wang does not disclose, expressly or inherently, that "said third code section is embedded within said second code section, and said second code section is embedded within said first code section," as is claimed in claims 7 and 26. While Wang does teach an original input source, such as HTML, that may include a plurality of embedded code

sections, such as embedded VB Script, Wang does not disclose a third code section that may be embedded in a second code section when the second is itself embedded in a first. The example given by Wang of a second code section (VB Script) embedded in a first code section (arguably, HTML) does make reference to the possibility of other additional code sections, but makes no explicit reference to such sections being embedded in VB Script or any other code section that might itself be embedded in HTML. Accordingly, Wang does not anticipate claims 7 and 26.

Examiner's response:

In response to Applicants argument, Wang discloses enabling multiple runtime processor executed by the computer. Each of the runtime processors process their respective intermediate sources derived from an original input source, i.e., Java or Visual Basic Script (See summary). In order to process multi language processor, Wang's system recognize different input source languages and invokes the respective processor according to the input source language (col. 2, lines 26-35). Further, Wang disclose synchronization token, which maps the language commands/code to the one need to, transferred i.e., first or second or third code sections (FIG. 2, and col. 4, line 64 to col. 5, line 8). Applicants make general allegations. Therefore, the rejection is proper and maintained herein.

c. Although Wang arguably makes reference to reading "a data processing representation having code sections with programming language statements of at least a first and a second programming language," no explicit mention is made of recognizing a

header section in at least one of the representation's programming language code sections. Nor is any "recognizing" operation inherent in Wang. The method and scripting environment of Wang merely requires that an HTML parser recognize code sections of different languages. No other recognition operations are required.

Examiner's response:

In response to Applicants argument, as understood by the specification (copied below for convenience) that the header section is nothing but the tag command within XML which is similar to the one disclosed by the reference Wang (FIG.2 and related discussion).

> In one embodiment, a header section **109** may be declared in accordance with the following exemplary syntax:
>
> ```
> <xs:header>
>     <java:directive>
>         import org.w3c.dom.*;
>     </java:directive>
> </xs:header>
> ```
>
> The above example directive directs the import of W3C's definition of the document object model for use by subsequent Java™ code sections.

Applicants make general allegations. Therefore, the rejection is proper and maintained herein.

d. Additionally, even assuming for the sake of argument some teaching to combine Wang and Claussen (Applicants do not concede that any such teaching exists), Claussen fails to disclose "wherein said first language is a directive language, and said second language is a selected one of XML and an object-oriented language," as recited by claim 4 (claims 5, 8, 23,24, and 27 contain similar recitations to the above mentioned recitation

of claim 4). Rather, Claussen teaches "a web page.., supporting multiple scripting languages is compiled into an XML... DOM (Document Object Model), and, thereafter, into a Java servlet." While multiple scripting languages are mentioned, Claussen simply does not suggest the use of a directive language, XML, or an object-oriented language as one or more of the multiple scripting languages. Instead, Claussen teaches the compiling of said languages into XML, and then into a Java servlet. Therefore, Claussen does not teach the limitation lacking in Wang, and thus, when combined with Wang, cannot form the basis for an obviousness rejection.

Examiner's response:

In response to Appellants argument, that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Claussen does disclose multiple scripting languages are supported by identifying a start and an end of each scripting language code block authored into the web page markup language which provides new techniques for publishing Internet content that can fully leverage the manipulation and template mechanism of XSLT with the scripting capability of the JSP/ASP model (col. 2, lines 51-55 and col. 2, lines 62-65). More specifically to the limitation, wherein said first language is a directive language, and said second language is a selected one of XML and an object-oriented language (col. 2-3, lines 66-67 and 1-2 "...supporting multiple

languages is compiled in to an XML... and thereafter, into a Java™ servlet..."). Further, Examiner has shown why it would have been obvious to incorporate the references to provide to provide a technique for publishing Internet content that can fully leverage the manipulation as suggested by Claussen (col. 2, lines 23-55) (see the rejection below). Applicants make general allegations. Therefore, the rejection is proper and maintained herein.

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claim 14-38 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding, claims 14, 17, 19, 33, 36, and 38 recited the limitation, "adapted to" is not clear whether the limitation following the adapted to are required or not. Therefore, the scope of the claims is vague and indefinite.

The rejection of the base claim is necessarily incorporated into the dependent claims.

10. Claims 14, 17, 19, 33, 36, and 38 recites the limitation "first and second code statements". There is insufficient antecedent basis for this limitation in the claim. Examiner interpreted the limitation "first and second code statements" as code statements.

The rejection of the base claim is necessarily incorporated into the dependent claims.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --
(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

12. Claims 1, 2, 3, 6, 7, 20, 22, 25, 26, 33, 36 and 38 are rejected under 35 U.S.C. 102(e) as being anticipated by 6,292,936 to Wang (hereinafter, Wang).

Per claims 1 and 2:

Wang discloses:

- A method of computing comprising:
- reading, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a data processing representation having code sections with code statements of at least a first and a second programming language (col. 1, lines 44-46 "Each of the runtime processors processes their respective corresponding intermediate sources derived from an original input source in a synchronous manner");
- recognizing, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a first code section with at least code statements of a first programming language (col. 2, lines 56-59 "The server system 106 may further include one or more translators 114 that are executed to translate the original input source for the runtime processors 110 and 112");
- invoking, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a first code statement processing unit of the first programming language to process the first code section (col. 1, lines 44-46 "Each of the runtime processors

processes their respective corresponding intermediate sources derived from an original input source in a synchronous manner");

- recognizing, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a second code section with at least code statements of a second programming language (col. 1, lines 46-48 "One or more of the respective corresponding intermediate sources includes a synchronizer token that provides synchronization among the runtime processors");
- invoking, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a second code statement processing unit of the second programming language to process the second code section (col. 1, lines 49-51 "Using the synchronizer token, an execution sequence of the original input source is maintained").

Per claim 3:

The rejection of claim 1 is incorporated, and further, Wang discloses:

- wherein said second code section is embedded within said first code section. The limitations in the claims are similar to those in claim 1, and rejected under the same rational set forth in connection with the rejection of claim 1.

Per claim 6:

The rejection of claim 1 is incorporated, and further, Wang discloses:

- recognizing a third code section with at least code statements of a third programming language (col. 2, lines 56-59 "The server system 106 may further include one or more translators 114 that are executed to translate the original input source for the runtime processors 110 and 112");
- invoking a third code statement processing unit of the third programming language to process the third code section (col. 1, lines 44-46 "Each of the runtime processors processes their respective corresponding intermediate sources derived from an original input source in a synchronous manner").

Per claim 7:

The rejection of claim 6 is incorporated, and further, Wang discloses:

- wherein said third code section is embedded within said second code section, and said second code section is embedded within said first code section. The limitations in the claims are similar to those in claim 6, and rejected under the same rational set forth in connection with the rejection of claim 6.

Claims 20, 21, 22, 25 and 26 are the apparatus claim corresponding to method claims 1, 3, 6 and 7 respectively, and rejected under the same rational set forth in connection with the rejection of claims 1, 3, 6 and 7 respectively, above, as noted above and Wang also discloses system, see FIG. 1 and associated text.

Claims 33, 36 and 38 are the apparatus claim corresponding to method claim 1, and rejected under the same rational set forth in connection with the rejection of claims 1, above, as noted above and Wang also discloses system, see FIG. 1 and associated text.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claims 4, 5, 8, 23, 24 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang in view of US Patent No. 6,732,330 to Claussen et al. (hereinafter, Claussen).

Per claim 4:

The rejection of claim 1 is incorporated, and further, Wang does not explicitly disclose wherein said first language is a directive language, and said second language is a selected one of XML and an object-oriented language.

However, Claussen discloses in an analogous computer system wherein said first language is a directive language, and said second language is a selected one of XML and an object-oriented language (col. 2-3, lines 66-67 and 1-2 "...supporting multiple languages is compiled in to an XML... and thereafter, into a Java™ servlet...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of wherein said first language is a directive language, and said second language is a selected one of XML and an object-oriented language

as taught by Claussen into the method of enabling multiple runtime processors in an embedded scripting system as taught by Wang. The modification would be obvious because of one of ordinary skill in the art would be motivated to use XML and Java to provide a technique for publishing Internet content that can fully leverage the manipulation as suggested by Claussen (col. 2, lines 23-55).

Per claim 5:

- wherein said first language is an object-oriented language, and said second language is XML. The limitations in the claims are similar to those in claim 4, and rejected under the same rational set forth in connection with the rejection of claim 4.

Per claim 8:

- wherein said first language is a directive language, said second language is an object-oriented language and said third language is XML. The limitations in the claims are similar to those in claim 4, and rejected under the same rational set forth in connection with the rejection of claim 4.

Claims 23, 24 and 27 are the apparatus claim corresponding to method claims 4, 5 and 8 respectively, and rejected under the same rational set forth in connection with the rejection of claims 4, 5 and 8 respectively, above, as noted above and Wang also discloses system, see FIG. 1 and associated text.

15. Claims 9-19, 28-32, 34, 34 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang in view of US Patent No. 5,428,792 to Conner et al. (hereinafter, Conner).

Per claim 9:

The rejection of claim 1 is incorporated, and further, Wang discloses:

- invoking the library function, and outputting the result of the invocation (col. 3, lines 40-42 "The remaining VisualBasic Script blocks in the original input source 116 are translated into notify method and wait method invocations").

Wang does not explicitly disclose wherein the method further comprises recognizing an invocation of a library function within at least a selected one of said first and second code sections.

However, Conner discloses in an analogous computer system wherein the method further comprises recognizing an invocation of a library function within at least a selected one of said first and second code sections (col. 7, lines 20-23 "class designer defines the class interface, implements the class methods, and finally loads the resulting object code into a class library").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of recognizing an invocation of a library function within at least a selected one of said first and second code sections as taught by Conner into the method of enabling multiple runtime processors in an embedded scripting system as taught by Wang. The modification would be obvious because of one of ordinary skill in the art

would be motivated to use a library function to provide the reusability of the OOP functions already exist as suggested by Conner (col. 1, lines 55-67).

Per claim 10:

The rejection of claim 1 is incorporated, and further, Wang does not explicitly disclose wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point.

However, Conner discloses in an analogous computer system wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point (col. 5, lines 1-12 "...class is a definition of an object... <stack> is an example of a class... stack contains two data elements (<stackArray> and <stackTop>), and supports three methods, <create()>, <push()>, and <pop()>...").

The feature of library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point would be obvious for the reasons set forth in the rejection of claim 9.

Per claim 11:

The rejection of claim 1 is incorporated, and further, Wang does not explicitly disclose wherein the method further comprises recognizing a header section of a selected one of the first and the second programming; recognizing a directive statement within the header section, enumerate one or more data packages; and importing the enumerated one or more data packages

for use within code sections with at least statements of the selected first and second programming language.

However, Conner discloses in an analogous computer system wherein the method further comprises recognizing a header section of a selected one of the first and the second programming language (col. 9, lines 35-40 "...a valid C header file which contains macros necessary to invoke public methods and access public data elements of the class... file... included in any client of the class, and is created by the SOM compiler"); recognizing a directive statement within the header section, enumerate one or more data packages (col. 25, lines 14-20 "section contains an include statement that is a directive to the OIDL preprocessor telling the compiler where to find the class interface definition for this class' parent class..."); and importing the enumerated one or more data packages for use within code sections with at least statements of the selected first and second programming language (col. 2, lines 19-21 "...bindings are input to the particular target language compiler to generate object module...").

The feature of recognizing a header section... recognizing a directive statement... and importing the enumerated... would be obvious for the reasons set forth in the rejection of claim 9.

Per claim 12:

- wherein the method further comprises recognizing a header section of a selected one of the first and the second programming language; recognizing a declare statement within the header section, enumerating one or more processing methods; and instantiating the enumerated one or more processing methods for use within code sections with at least

statements of the selected first and second programming language. The limitations in the claims are similar to those in claim 11, and rejected under the same rational set forth in connection with the rejection of claim 11.

Per claim 13:

- wherein the method further comprises recognizing a header section of a selected one of the first and the second programming language; recognizing a declare statement within the header section, enumerating one or more instance variables; and instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language. The limitations in the claims are similar to those in claim 11, and rejected under the same rational set forth in connection with the rejection of claim 11.

Per claim 14:

Wang disclose:

- A method of computing comprising:
- reading, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a data processing representation having code sections with code statements of at least a first and a second programming language (col. 1, lines 44-46 "Each of the runtime processors processes their respective corresponding intermediate sources derived

from an original input source in a synchronous manner"), the execution engine adapted to recognize the code sections and, in response, invoke first and second statement processing units to process the code sections (col. 3, lines 24-29 "HTML Parser 114 translates the original input source 116 and generates two intermediate sources, e.g., one 200 for the Java Virtual Machine 110 and one 202 for the VisualBasic Script interpreter 112").

Wang does not explicitly disclose recognizing, by an execution engine, a header section of a selected one of the first and the second programming language; recognizing, by an execution engine, a directive statement within the header section, enumerating one or more data packages; and importing, by an execution engine, the enumerated one or more data packages for use by code sections within code sections with at least statements of the selected first and second programming language.

However, Conner discloses in an analogous computer system recognizing, by an execution engine, a header section of a selected one of the first and the second programming language (col. 9, lines 35-40 "...a valid C header file which contains macros necessary to invoke public methods and access public data elements of the class... file... included in any client of the class, and is created by the SOM compiler"); recognizing, by an execution engine, a directive statement within the header section, enumerating one or more data packages (col. 25, lines 14-20 "section contains an include statement that is a directive to the OIDL preprocessor telling the compiler where to find the class interface definition for this class' parent class..."); and importing, by an execution engine, the enumerated one or more data packages for use by code

sections within code sections with at least statements of the selected first and second programming language (col. 2, lines 19-21 "...bindings are input to the particular target language compiler to generate object module...").

The feature of recognizing, by an execution engine, a header section... recognizing a directive statement... and importing, by an execution engine, the enumerated... would be obvious for the reasons set forth in the rejection of claim 9.

Per claim 15:

- wherein the method further comprises recognizing a declare statement within the header section, enumerating one or more processing methods; and instantiating the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language. The limitations in the claims are similar to those in claim 14, and rejected under the same rational set forth in connection with the rejection of claim 14.

Per claim 16:

- wherein the method further comprises recognizing a declare statement within the header section, enumerating one or more instance variables; and instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language. The limitations in the claims are similar to those in claim 14, and rejected under the same rational set forth in connection with the

rejection of claim 14.

Per claim 17:

Wang disclose:

- A method of computing comprising:
- reading, by an execution engine(col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a data processing representation having code sections with code statements of at least a first and a second programming language (col. 1, lines 44-46 "Each of the runtime processors processes their respective corresponding intermediate sources derived from an original input source in a synchronous manner"), the execution engine adapted to recognize the code sections and, in response, invoke first and second statement processing units to process the code sections (col. 3, lines 24-29 "HTML Parser 114 translates the original input source 116 and generates two intermediate sources, e.g., one 200 for the Java Virtual Machine 110 and one 202 for the VisualBasic Script interpreter 112").

Wang does not explicitly disclose recognizing, by an execution engine, a header section of a selected one of the first and the second programming language; recognizing, by an execution engine, a first declare statement within the header section, enumerating one or more processing methods; and instantiating, by an execution engine, the enumerated one or more processing

methods for use within code sections with at least statements of the selected first and second programming language.

However, Conner discloses in an analogous computer system recognizing, by an execution engine, a header section of a selected one of the first and the second programming language (col. 9, lines 35-40 "...a valid C header file which contains macros necessary to invoke public methods and access public data elements of the class... file... included in any client of the class, and is created by the SOM compiler"); recognizing, by an execution engine, a first declare statement within the header section, enumerating one or more processing methods (col. 25, lines 14-20 "section contains an include statement that is a directive to the OIDL preprocessor telling the compiler where to find the class interface definition for this class' parent class..."); and instantiating, by an execution engine, the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language (col. 2, lines 19-21 "...bindings are input to the particular target language compiler to generate object module...").

The feature of recognizing a header section... recognizing a directive statement... an d importing the enumerated... would be obvious for the reasons set forth in the rejection of claim 9.

Per claim 18:

- wherein the method further comprises recognizing a second declare statement within the header section, enumerating one or more instance variables; and instantiating the enumerated one or more instance variables for use within code sections with at least

statements of the selected first and second programming language. The limitations in the claims are similar to those in claim 17, and rejected under the same rational set forth in connection with the rejection of claim 17.

Per claim 19:

Wand discloses:

- A method of computing comprising:
- reading, by an execution engine (col. 2, lines 49-55 "...run time processors...that are executed...comprises a Java Virtual Machine...executes Java programming statements...VisualBasic Script interpreter... executes VisualBasic Script programming statements"), a data processing representation having code sections with code statements of at least a first and a second programming language (col. 1, lines 44-46 "Each of the runtime processors processes their respective corresponding intermediate sources derived from an original input source in a synchronous manner"), the execution engine adapted to recognize the code sections and, in response, invoke first and second statement processing units to process the code sections (col. 3, lines 24-29 "HTML Parser 114 translates the original input source 116 and generates two intermediate sources, e.g., one 200 for the Java Virtual Machine 110 and one 202 for the VisualBasic Script interpreter 112").

Wang does not explicitly disclose recognizing, by an execution engine, a header section of a selected one of the first and the second programming language; recognizing, by an execution

engine, a declare statement within the header section, enumerating one or more instance variables; and instantiating, by an execution engine, the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

However, Conner discloses in an analogous computer system recognizing, by an execution engine, a header section of a selected one of the first and the second programming language (col. 9, lines 35-40 "...a valid C header file which contains macros necessary to invoke public methods and access public data elements of the class... file... included in any client of the class, and is created by the SOM compiler"); recognizing, by an execution engine, a declare statement within the header section, enumerating one or more instance variables (col. 25, lines 14-20 "section contains an include statement that is a directive to the OIDL preprocessor telling the compiler where to find the class interface definition for this class' parent class..."); and instantiating, by an execution engine, the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language (col. 2, lines 19-21 "...bindings are input to the particular target language compiler to generate object module...").

The feature of recognizing a header section... recognizing a directive statement... and importing the enumerated... would be obvious for the reasons set forth in the rejection of claim 9.

Claims 28-32 are the apparatus claim corresponding to method claims 9-13 respectively, and rejected under the same rational set forth in connection with the rejection of claims 9-13

respectively, above, as noted above and Wang also discloses system, see FIG. 1 and associated text.

Claims 34, 35 and 37 are the apparatus claim corresponding to method claim 13, and rejected under the same rational set forth in connection with the rejection of claim 13, above, as noted above and Wang also discloses system, see FIG. 1 and associated text.

Conclusion

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571) 272-3732**. The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to Friday except every other Friday and federal holidays. Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: 571-272-2100**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Wei Y. Zhen** can be reached on **(571) 272-3708**. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner/Software Engineer
Art Unit 2191

WEI ZHEN
SUPERVISORY PATENT EXAMINER